REMARKS

In paragraph 1 of the Office Action, the Examiner notes that the Draftsperson objected to three of the original drawings (Figures 2, 3, and 5) in this application as having improper left margins. In response, three new formal drawing sheets with proper left margins are presented as replacements to overcome the objections.

An error in the originally-submitted oath in this application has been discovered and corrected by a supplemental oath submitted with this response, per MPEP 602.07 and 37 CFR 1.67(a)(2). Inventor Narayana Iyer Subramanian was misidentified in the original oath as Narayana Iyer.

REJECTIONS UNDER 35 U.S.C. 102(b)

In paragraph 3 of the Office Action, the Examiner rejected claims 1-13, 17-34, and 38-44 as anticipated by Monday. Applicants respectfully traverse these rejections.

A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference. See MPEP 2131. Applicants assert this standard has not been met by the cited prior art.

Regarding claim 1, for example, each claimed element is supported by at least the following portions of the specification:

- automatically mapping each table in a relational database to a virtual XML document (the Default View), page 9, line 20, page 10, lines 1-7, Fig 3.

- using an XML query over the Default View as the means to extract and mark up relational data. Tables in the underlying database are queried in an XML query language as if they were XML documents, page 10, lines 9-17, Fig 4.

- transforming XML queries into an "intermediate representation", which describes how the underlying relational data should be selected, marked up, grouped, and nested to generate the desired XML document, page 10, lines 19-21, page 11, lines 1-21, Fig 5.

- rewriting the intermediate representation into a) a query part that can be executed as a SQL query and b) a tagger part that can be executed by a tagging engine, page 13, lines 5-20, Fig 6, Fig 7.

- translating the rewritten intermediate representation into SQL query and tagger instructions, page 14, lines 8-20, Fig 8.

- executing the SQL query and its streaming results into a tagging engine, which produces the resulting XML document, page 15, lines 6-20, page 16, lines 1-12, Fig 9.

Referring now to the rejection of the independent claims in paragraph 4 of the Office Action, Applicants provide a point-by-point refutation of the Examiner's assertions:

- "mapping a number of relational database tables to a number of virtual XML documents". Monday teaches a Java Bean being used to access the underlying database. There is no explicit or implied notion of mapping a relational database to a virtual XML document that can then be used in an XML query, as described in the present application.

- "issuing XML queries over said virtual XML documents" and "parsing said XML queries". Monday says nothing at all about how an XML query language could be used as the means to extract data from a relational database, as described in the present application.

- "It transforming (sic) said XML queries into a language-neutral intermediate representation". Monday states only that there is a "software program that bridges the gap between the markup language interface and database" (col 5, lines 25-31). Monday says nothing about an intermediate representation like the one described in the present application, with BIND, GROUP, CONSTRUCT, etc.

- "rewriting said language-neutral intermediate representation into an equivalent form easily translated into a SQL query" and "translating said equivalent form into a SQL query over said relational database tables and into tagging instructions passed to a tagger" and "executing said SQL query to produce SQL query results passed to said tagger" and "generating XML output using said SQL query results and said tagging instructions". Monday says nothing at all about an intermediate representation of an XML query, how to generate SQL from an intermediate representation, or a tagger that can take a stream of rows from said SQL query and generate a result XML document. In contrast, the specification of the present invention describes a scheme to rewrite the intermediate representation into a) a query part that can be executed as a SQL query and b) a tagger part that can be executed by a tagging engine.

Also, Monday's method will only work for single data objects, like Java Beans (Fig 2 and Fig 3). Their Data Request method is not general enough to work on relational tables, where each row has to be selected, marked up, grouped, and nested. In particular, the algorithm shown in Fig 4 cannot select the rows of a table, individually mark them, then group and nest them within the marked up rows of another table. In contrast, the present application describes an intermediate representation, along with rewrite and translation algorithms for that intermediate representation to deal with this problem.

In general, database access in Monday is made explicit with a "Data Request", which is basically an annotated DTD (see Fig 6, col 8, lines 47-65 of Monday). There is no notion a Default View, where the tables of a relational database can be viewed by users as virtual

XML documents that can be queried with an XML query language. Further, Monday's Data Request method relies on data objects like Java Beans to access the database (see Figs 2-3, col 7, lines 42-67 of Monday). An XML query language is not used to as the means to request data. Hence, Monday does not describe or anticipate a method to transform an XML query into an intermediate representation, which is in turn rewritten and translated into a SQL query and tagger instructions.

REJECTIONS UNDER 35 U.S.C. 103(a)

Claims 14-16 and 35-37 stand rejected under 35 U.S.C. 103(a) as unpatentable over Monday in view of Chen. Applicants respectfully traverse these rejections. Remarks above regarding Monday also apply to these claims.

Chen describes a merge algorithm, which takes an input XML document in one format (the "input DTD"), a format specification for return XML documents (the "return DTD"), and a "name tag map". The merge algorithm uses these to transform an input XML document into a return XML document (see Fig 7, col 4, lines 40-54, Fig 11A-11E, col 8, lines 20-35 of Chen). However, the merge algorithm in Chen does not use an XML query language as the means to request and markup relational data. Hence, Chen neither teaches nor suggests transforming an XML query into an intermediate representation, which is in turn rewritten and translated into a SQL query and tagger instructions. Further, Chen starts with a marked up input XML document, not unmarked, raw relational data. Thus, Chen neither teaches nor suggests a method that can markup data from relational tables, where each row of a table is marked up, grouped and nested as described in the present application.

Regarding the rejections in paragraphs 24 and 26 of the Office action, while it is true that Chen describes how to group and nest, this is only in the context of transforming an input XML document to a return XML document, while working with a return DTD and name table map. Chen neither teaches nor suggests how to group and nest raw relational data from a SQL query, as in the present invention, and the addition of Monday does not remedy this shortcoming.

Regarding the rejections in paragraph 25 of the Office Action, Monday and Chen say nothing about an intermediate representation like the one described in the present application, with BIND, GROUP, CONSTRUCT, etc. They also say nothing about rewrite rules to transform an intermediate representation into a) a query part that can be executed as a SQL query and b) a tagger part that can be executed by a tagging engine. The intermediate representation and rewrite rules of the present invention are particular to the problem of how to select, mark up, group, and nest raw relational data while using an XML query language as the means to request and generate the desired resulting XML document, and are neither taught nor suggested, alone or in combination by the cited prior art references. Monday only deals with "data objects" as input data (like Java Beans). Chen deals only with XML as input data. Neither Monday nor Chen discuss the use of an XML query language for requesting and generating the desired XML document.

All pending claims are believed to be allowable. The prior art made of record and not relied upon has been carefully considered. The Examiner is invited to call Applicants' undersigned representative if a telephone conference will expedite the prosecution of this application.

Respectfully submitted,

M. Carey et al.

By Marc D. McSwain

Marc D. McSwain (#44,929)
Phone (408) 927-3364